BIOINFORMATICS APPLICATIONS NOTE Vol. 23 no. 15 2007, pages 2024–2027
doi:10.1093/bioinformatics/btm290

Data and text mining

VISDA: an open-source caBIG™ analytical tool for data clustering and beyond

Jiajing Wang[1], Huai Li[2], Yitan Zhu[1], Malik Yousef[3], Michael Nebozhyn[3], Michael Showe[3], Louise Showe[3], Jianhua Xuan[1], Robert Clarke[4] and Yue Wang[1,*]

[1]Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Arlington, VA 22203, [2]Bioinformatics Unit, RRB, National Institute on Aging, NIH, Baltimore, MD 21224, [3]Systems Biology Division, The Wistar Institute, Philadelphia, PA 19104 and [4]Lombardi Comprehensive Cancer Center and Department of Oncology, Georgetown University, Washington, DC 20057, USA

Received on November 8, 2006; revised on May 22, 2007; accepted on May 22, 2007

Advance Access publication May 31, 2007

Associate Editor: John Quackenbush

ABSTRACT

Summary: VISDA (Visual Statistical Data Analyzer) is a caBIG™ analytical tool for cluster modeling, visualization and discovery that has met silver-level compatibility under the caBIG initiative. Being statistically principled and visually interfaced, VISDA exploits both hierarchical statistics modeling and human gift for pattern recognition to allow a progressive yet interactive discovery of hidden clusters within high dimensional and complex biomedical datasets. The distinctive features of VISDA are particularly useful for users across the cancer research and broader research communities to analyze complex biological data.

Availability: http://gforge.nci.nih.gov/projects/visda/

Contact: yuewang@vt.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Many biomedical hypothesis-driven studies can be formulated as a clustering problem, such as classifying cancer subtypes, detecting data outliers and identifying functional gene modules (de Hoon *et al*., 2004). Clustering a complex dataset within high dimensions is a challenging task. Several tools are currently available to cluster and display complex biological data (Dudoit *et al*., 2003; Gentleman *et al*., 2004; Reich *et al*., 2006). Naturally, more capable methods are continuously being developed for the analysis and interpretation of complex datasets that may include multiple subclasses.

VISDA is an open-source clustering tool developed to target the silver-level requirements of the cancer biomedical informatics grid (caBIG™) architecture and compatibility guidelines (https://cabig.nci.nih.gov/tools/VISDA). caBIG™ is a major initiative of the National Cancer Institute to create an open-source, open-access information network enabling cancer researchers to share tools, data, applications and technologies according to agreed-upon standards and identified needs. Compared to existing methods such as hierarchical clustering (HC) and self-organizing map (SOM) as provided by many popular analytical tools (Dudoit *et al*., 2003; Eisen *et al*., 1998; Gentleman *et al*., 2004; Tamayo *et al*., 1999), VISDA produces a coarse-to-fine cluster structure supported by a statistical hierarchical mixture model and supervised/unsupervised feature selection to ease the curse of dimensionality. The embedded hierarchical data exploration scheme helps discover and visualize the hidden tree of clusters. Interactive user participation directs the clustering process; the clustering solution is validated by a minimum description length (MDL) based model selection. Finally, VISDA uses soft data decomposition to model overlapped clusters (Wang *et al*., 2000;Wang *et al*., 2003). Thus, VISDA offers both an adjunct and an alternative to existing methods.

As one of the adopted data analytical tools in caBIG™, VISDA offers users across the cancer research and broader research communities a unique yet effective data clustering tool for cluster modeling, discovery and visualization. VISDA is an open-source software. The Java and C source code and documents of application program interfaces (APIs) are provided at http://gforge.nci.nih.gov/projects/visda/ enabling users to modify the program and add new functions or extensions.

2 DESCRIPTION

2.1 Dependencies

The core algorithms of VISDA are implemented in C++ and the visualization functions and user interface are implemented in Java. To improve the design performance and speed up the design time, we used the Linear Algebra Package (LAPACK) for basic vector and matrix operations and two open Java packages, jmathplot and epsgraphics for plotting graphics. Importing data from the caArray data portal to VISDA is facilitated by caArray MAGE-OM APIs. Importing data from local MAGE-ML file to VISDA uses MAGEstk (The MAGE Software Toolkit). VISDA has been tested on Microsoft Windows XP, Linux and UNIX platforms. Users can install VISDA directly on a computer and launch

*To whom correspondence should be addressed.

the program from batch files provided in the deployment package.

2.2 Architectural design

VISDA contains four major components: (1) edu.vt.cbil.visda, (2) edu.vt.cbil.visda.data, (3) edu.vt.cbil.visda.comp and (4) edu.vt.cbil.visda.view. The class objects and their relations are designed using Unified Modeling Language (UML). edu.vt.cbil.visda package provides a main entry to perform the initial setup, data input and output, data analysis and data/results visualization by calling well-defined APIs. edu.vt.cbil.visda.data can interact with microarray data sources such as caArray database, local MAGE-ML files and tab-delimited text files and then make the data available to edu.vt.cbil.visda.comp. In edu.vt.cbil.visda.comp, we implemented the following modules: (1) cluster modeling module (CMM), (2) dimension reduction module (DRM), (3) cluster formation module (CFM), and (4) cluster validation module (CVM). These modules comprise the essential cores of the VISDA toolkit (Wang *et al*., 2000). edu.vt.cbil.visda.view can display the data profile, and both the intermediate and final output results including the 'soft-clustering' probabilities of the samples/genes in each cluster and a graphical representation of the estimated hierarchical 'tree of phenotype' and/or 'tree of gene module'.

2.3 Application features

2.3.1 Data preprocessing The input data can be (i) any tab-delimited text file including multiple annotations of genes and conditions; (ii) any local data file in MAGE-ML format and (iii) data retrieved from caArray. All gene/sample annotation fields can be automatically extracted and used for subsequent cluster discovery. The uploaded data can also be visualized as a heatmap before the scheduled analysis, providing the user a global view of the entire data set. The configuration step gives a user the freedom to choose among different analysis tasks, such as gene/phenotype clustering, supervised/unsupervised feature selection, various projection methods and other advanced features including cluster validation. VISDA core algorithms are then activated to perform the targeted clustering on the uploaded gene expression data.

2.3.2 Analytical algorithms VISDA implements the following functions for sample/gene clustering: (i) supervised and unsupervised feature selections; (ii) discriminatory data projections for exploratory cluster visualization, including principal component analysis (PCA) and projection pursuit method (PPM); (iii) hierarchical statistical modeling and parameter estimation by the expectation-maximization (EM) algorithm and (iv) advanced functional options including Fisher discriminatory component analysis (DCA) projection, MDL cluster validation and hybrid clustering initialization using HC-k-means/SOM.

2.3.3 Information visualization Expression data are displayed as a heatmap. Annotations of the conditions are shown at the top; annotations of the genes are listed on the right. During the clustering process, clusters at each hierarchical level can be visualized by three individual 2D projections: PCA, PPM and DCA. The user can then select the best projection view for further classification at the deeper–levels. One of VISDA's distinctive features is the integration of human intelligence into the automation of the core algorithms. To leverage a user's prior knowledge and visual cues about data patterns, VISDA allows each user to initialize the number of clusters and their centers at each exploration level. The iterative user-algorithm interactions exploit the power of the human gift for pattern recognition and statistical machine learning, assuring robust and globally converged clustering solutions.

2.3.4 Graphic user interface—GUI Figure 1 shows representative screen shots from VISDA. All the sub-level results are stored into a hierarchical structure, and a pie chart diagram (Fig. 1D) shows the growth of the HC tree. All the pictures can be viewed, zoomed and saved in either PNG or EPS format. At each hierarchical level, the clustering posterior probabilities of all samples/genes belonging to each cluster can be saved as a text table with multiple sample/gene annotations. The table of the most informative genes/features selected for array clustering, ranked by their respective signal-to-noise ratio (SNR; supervised) or variance (unsupervised) criteria, can also be viewed and saved.

2.3.5 Case study VISDA has been tested and used in several ongoing projects for cancer diagnostic and muscular dystrophy studies and shown its effectiveness for subclass discovery and novel cluster detection (Bakay *et al*., 2006; Zhu *et al*., 2006). The Showe's lab at the Wistar Institute (Philadelphia, PA, USA) applied VISDA to a dataset of two head and neck cancer subtypes. VISDA reveals two novel subclusters in one of the subtypes, providing new biological insights into cancer development. Guided by a pathologically plausible diagnostic Tree of Phenotype (TOP), we conducted gene clustering by VISDA on a microarray gene expression dataset of 12 different muscle dystrophies and normal skeletal muscle. We then superimposed prior knowledge of gene regulation to analyze the clustering results and generate novel hypotheses for further research on muscular dystrophies. We obtained several condition-specific gene bi-clusters at different nodes/levels of the VISDA derived TOP, and shown their potential association with specific pathway gene regulatory networks (Zhu *et al*., 2006).

3 DISCUSSION

VISDA was developed to be consistent with the caBIG™ silver compatibility guidelines that highlight the use of controlled vocabularies, common data elements (CDEs), well-documented APIs and UML models. As a caBIG™ analytical tool, VISDA is capable to retrieve data from caArray (caGrid node) and locally perform computations upon these data. We implemented VISDA client APIs to consume silver-compatible MAGE-OM APIs and caArray CDEs. Class diagrams in UML were provided for all VISDA packages and APIs. Documentations of the programming interfaces were derived from the implementation using JavaDocs.

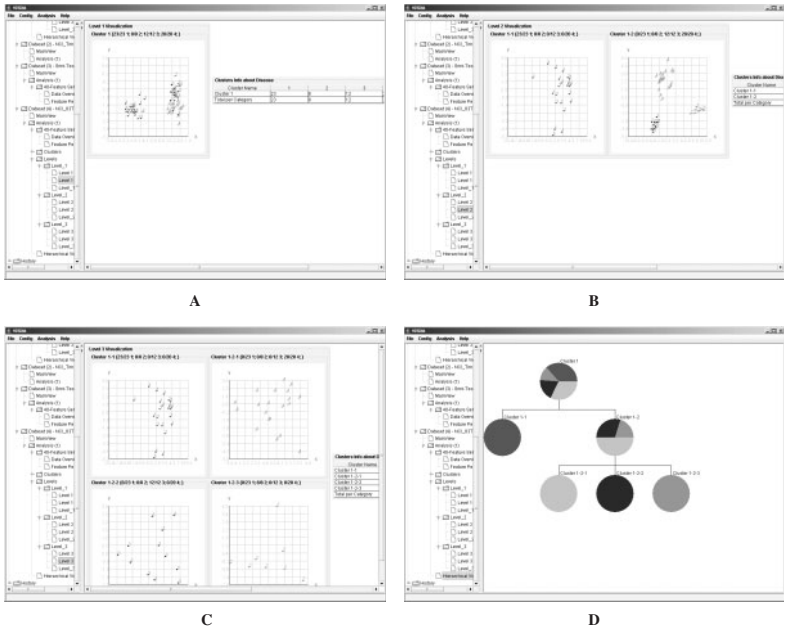

Fig. 1. Representative screen shots from VISDA for a top–down clustering of a real data set with ground truth (color coded). A navigation panel on the left leads users through the process of data analysis. A working view panel on the right displays the analysis results. (**A**) A projection view at level 1; (**B**) a projection view at level 2; (**C**) a projection view at level 3 (**D**) a hierarchical visualization of the clustering and outcomes. Color version of this figure is available as Supplementary material online.

There are several workflow scenarios for using caBIG™ VISDA. For example, end users can access the data files and annotations through the caArray data portal and perform clustering analysis. End users can also access the data files and annotations from caArray, then conduct normalization by distance weighted discrimination (DWD, another caBIG™ analytical tool), or by other normalization tools provided in caBioconductor (an ongoing caBIG™ project), and finally perform clustering analysis by VISDA. Using the current version of VISDA, users can retrieve a BioDataCube for DerivedBioArray of the selected experiment as well as associated composite sequence names and sample names via the MAGE-OM API, and then generate data matrix for all the arrays. Composite sequence names are obtained from DesignElement objects. caArray 1.3 or above system automatically annotates the sample name in the biomaterial annotation, delimited by '_L_'. Therefore, VISDA labels samples by parsing biomaterial names in BioMaterialMeasurement objects from the BioAssay. We noticed that some BioAssay objects retrieved from caArray do not contain BioMaterialMeasurement objects. In such cases, VISDA assigns bioassay names as sample names.

Human data interaction by VISDA can easily encode domain knowledge when used by domain experts. The statistical tree of clusters revealed by VISDA may provide meaningful relational biological information, and also allow cluster analysis at multiple resolutions. Since clustering algorithms always reflect some structural bias associated with the involved grouping principle (Frey and Dueck, 2007), it is recommended that for a new dataset without much prior knowledge one should try several different clustering methods or use an ensemble scheme that combines the results of different algorithms.

We plan to extend VISDA to include additional advanced analytical functions, such as unsupervised or semi-supervised gene/feature selection, outlier detection and conditional clustering via iterative and combinatorial gene and sample clustering. Other potential additions include using a stability analysis-guided phenotype clustering and visualization method to discover a highly resolved TOP from genomic data. Eventually, we will integrate VISDA into caGrid as a full functioned analytical service component.

ACKNOWLEDGEMENTS

The authors would like to thank members of the caBIG™ Integrative Cancer Research WorkSpace for reviewing VISDA documentation and/or providing helpful feedback on VISDA development. Thanks also go to the caArray Team at the National Cancer Institute for providing caArray testing datasets. This work is supported by the National Cancer Institute of the NIH under Grants caBIG™-VISDA; CA109872; CA100970; CA096483.

Conflict of Interest: none declared.

REFERENCES

Bakay,M. *et al.* (2006) Nuclear envelope dystrophies show a transcriptional fingerprint suggesting disruption of Rb-MyoD pathways in muscle regeneration. *Brain*, **129**, 996–1013.

de Hoon,M.J. *et al*. (2004) Open source clustering software. *Bioinformatics*, **20**, 1453–1454.

Dudoit,S. *et al*. (2003) Open source software for the analysis of microarray data. *Biotechniques*, Suppl. 45–51.

Eisen,M.B. *et al*. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, **95**, 14863–14868.

Frey,B.J. and Dueck,D. (2007) Clustering by passing messages between data points. *Science*, **315**, 972–976.

Gentleman,R.C. *et al*. (2004) Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol*., **5**, R80.

Reich,M. *et al*. (2006) GenePattern 2.0. *Nat. Genet*., **38**, 500–501.

Tamayo,P. *et al*. (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl Acad. Sci. USA*, **96**, 2907–2912.

Wang,Y. *et al*. (2000) Probabilistic principal component subspaces: a hierarchical finite mixture model for data visualization. *IEEE Trans. Neural Netw*., **11**, 625–636.

Wang,Z. *et al*. (2003) Discriminatory mining of gene expression microarray data. *J. VLSI Signal Process*., **35**, 255–272.

Zhu,Y. *et al*. (2006) Phenotypic-specific gene clustering using diagnostic tree and VISDA. *Proceedings of the 28th IEEE EMBS Annual International Conference*, 5767–5770.